

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-283251

(P2001-283251A)

(43) 公開日 平成13年10月12日 (2001. 10. 12)

(51) Int.Cl.⁷

G 0 6 T 15/00

識別記号

3 0 0

1 0 0

F I

G 0 6 T 15/00

データベース* (参考)

3 0 0

1 0 0 A

審査請求 未請求 請求項の数17 O L (全 19 頁)

(21) 出願番号 特願2001-52884(P2001-52884)

(22) 出願日 平成13年2月27日 (2001. 2. 27)

(31) 優先権主張番号 0 9 / 5 1 4 5 4 6

(32) 優先日 平成12年2月28日 (2000. 2. 28)

(33) 優先権主張国 米国 (US)

(71) 出願人 597067574

ミツビシ・エレクトリック・リサーチ・ラ
ボラトリーズ・インコーポレイテッド
アメリカ合衆国、マサチューセッツ州、ケ
ンブリッジ、ブロードウェイ 201

(72) 発明者 ハンスピーター・フィスター

アメリカ合衆国、マサチューセッツ州、サ
マービル、パーク・ストリート 60

(72) 発明者 マチアス・ペー・ツヴィッカー

スイス国、8500 ゲルリコン、イム・ヴェ
ーベルリスレプベルク 8

(74) 代理人 100057874

弁理士 曾我 道照 (外4名)

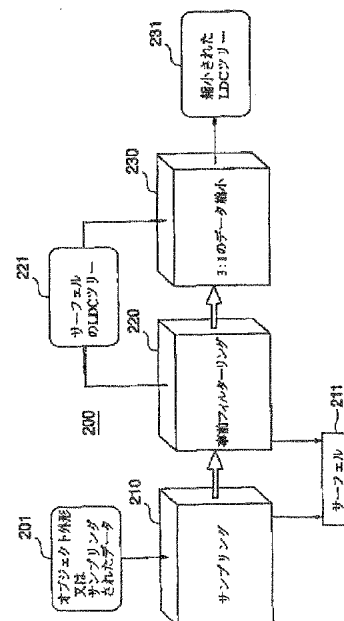
最終頁に続く

(54) 【発明の名称】 グラフィックオブジェクトをレンダリングする方法及び装置

(57) 【要約】

【課題】 どのようなオブジェクトにおいても、表面要素に変換することができるグラフィックオブジェクトをレンダリングする方法及び装置を得る。

【解決手段】 バイブラインは、オブジェクトの表面の形状と陰属性を記憶するメモリを有する。属性は、メモリ内のオクトリに配列される。オクトリは、複数のレベルで配列されたノードを有し、各ノードが、複数のゼロ次元のn個の要素からなる集合を記憶し、集合は、グラフィックオブジェクトの表面の一部分の形状及び陰属性を局部的に近似させ、n個の要素から成る集合は、画像空間のサンプリング解像度を有する。複数の並列処理バイブラインは、メモリに結合されている。バイブラインは、オクトリのノードのn個の要素から成る集合を最低解像度レベルから最高解像度レベルまでトラバースさせることで選択された向きを有した画像面にオクトリの形状及び陰属性を投影する。



【特許請求の範囲】

【請求項1】 グラフィックオブジェクトをレンダリングする方法であって、メモリに記憶されているオクトリにおけるオブジェクトの表面の形状及び陰属性を表す工程であって、上記オクトリは、複数のレベルで配列された複数のノードを有し、各ノードは、複数のゼロ次元の n 個の要素から成る集合を記憶し、各 n 個の要素から成る集合が、グラフィックオブジェクトの表面の一部分の形状及び陰属性を局部的に近似させ、 n 個の要素から成る集合は、画像空間のサンプリング解像度を有する、工程と、複数の並列レンダリングパイプラインを使用して画像面にオクトリの形状及び陰情報を階層的に投影する工程であって、該画像面は、オクトリのノードの n 個の要素から成る集合を最低解像度レベルから最高解像度レベルまでトラバースさせることで選択された向きを有する、工程とを有することを特徴とするグラフィックオブジェクトをレンダリングする方法。

【請求項2】 上記オクトリにおけるオブジェクトの形状及び陰属性を表すためにユニプロセッサでソフトウェアのプログラムを実行する工程と、上記メモリに結合された複数の並列処理パイプラインに形状及び陰属性を投影する工程とをさらに有することを特徴とする請求項1に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項3】 上記グラフィックオブジェクトは、上記画像面の解像度に従って適応できるようにサンプリングされることを特徴とする請求項1に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項4】 オブジェクト空間における n 個の要素から成る集合を、任意に選択された向きを有する上記画像面にワープする工程をさらに有することを特徴とする請求項1に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項5】 各 n 個の要素から成る集合のオクトリは、独立してワープされることを特徴とする請求項4に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項6】 特定の n 個の要素から成る集合を、深度バッファの対応するピクセル上に投影する工程と、投影された n 個の要素から成る集合の深度値が、ピクセルの以前に記憶された深度値よりも小さい場合にのみ、投影された n 個の要素から成る集合の深度値をピクセルに記憶する工程と、正接円盤を対応する n 個の要素から成る集合の或る位置に構築する工程であって、上記正接円盤は、 n 個の要素から成る集合間の最大距離よりも大きな半径を有する、工程と、上記正接円盤をピクセルの対応する部分集合上に投影する工程と、

投影された正接円盤の深度値が、ピクセルの対応する部分集合の深度値よりも小さい場合にのみ、投影された正接円盤の深度値をピクセルの対応する部分集合に記憶する工程とをさらに有することを特徴とする請求項1に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項7】 上記形状の属性は、複数のテクスチャマップを有し、 n 個の要素から成る集合の色を決定するために上記複数のテクスチャマップを補間する工程をさらに有することを特徴とする請求項1に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項8】 $\sqrt{2s/2}$ に等しい半径を有したフィルタを使用して色をフィルタリングする工程であって、但し、 s は、 n 個の要素から成る集合間の距離である、工程をさらに有することを特徴とする請求項7に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項9】 上記フィルタリングは、投影された n 個の要素から成る集合の密度に依存することを特徴とする請求項8に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項10】 上記深度バッファの解像度は、画像面の解像度よりも大きいことを特徴とする請求項6に記載のグラフィックオブジェクトをレンダリングする方法。

【請求項11】 グラフィックオブジェクトをレンダリングする装置であって、オブジェクトの表面の形状及び陰属性を記憶するメモリであって、該属性は、メモリにオクトリとして配列され、該オクトリは、複数のレベルで配列された複数のノードを有し、該各ノードは、複数のゼロ次元の n 個の要素から成る集合を記憶し、各 n 個の要素から成る集合は、グラフィックオブジェクトの表面の一部分の形状及び陰属性を局部的に近似させ、 n 個の要素から成る集合は、画像空間のサンプリング解像度を有する、メモリと、上記メモリに結合された複数の並列処理パイプラインであって、該処理パイプラインは、上記オクトリの形状及び陰の情報を画像面に投影し、画像面は、上記オクトリのノードの n 個の要素から成る集合を最低解像度レベルから最高解像度レベルまでトラバースさせることで選択された向きを有する、パイプラインとを備えたことを特徴とするグラフィックオブジェクトをレンダリングする装置。

【請求項12】 各ノードは、ブロックとして記憶され、各パイプラインは、ブロック選択ステージと、フォワードワーピングステージと、可視性ブラッシングステージと、テクスチャフィルタリングステージと、

視野依存シェーディングステージと、画像再構築及びアンチエイリアシングステージとを備えたことを特徴とする請求項11に記載のグラフィックオブジェクトをレンダリングする装置。

【請求項13】 上記ブロック選択ステージは、可視円錐を利用することを特徴とする請求項12に記載のグラフィックオブジェクトをレンダリングする装置。

【請求項14】 上記フォワード順ワーピングステージは、オブジェクト空間におけるn個の要素から成る集合を任意に選択された向きを有した画像面に対してワーブ

することを特徴とする請求項12に記載のグラフィックオブジェクトをレンダリングする装置。

【請求項15】 各n個の要素から成る集合のオクトリは、独立してワーブされることを特徴とする請求項12に記載のグラフィックオブジェクトをレンダリングする装置。

【請求項16】 深度バッファをさらに有し、特定のn個の要素から成る集合は、上記深度バッファの対応するピクセル上に投影され、投影されたn個の要素から成る集合の深度値は、投影されたn個の要素から成る集合の深度値が、ピクセルの以前に記憶された深度値よりも小さい場合にのみピクセルに記憶されることを特徴とする請求項11に記載のグラフィックオブジェクトをレンダリングする装置。

【請求項17】 上記陰属性は、複数のテクスチャマップを有し、テクスチャフィルタリングステージは、n個の要素から成る集合の色を決定するために複数の上記テクスチャマップを補間することを特徴とする請求項12に記載のグラフィックオブジェクトをレンダリングする装置。

【発明の詳細な説明】

【0001】
【発明の属する技術分野】この発明は、一般に、グラフィックレンダリングに関し、より具体的にはパイプライン接続されたレンダリングエンジンを使用してグラフィックオブジェクトの0次元表面要素をレンダリングする技術に関する。

【0002】

【従来の技術】3次元のコンピュータグラフィックスは、消費者レベルではいたるところで見受けられるようになっている。高級なPCワークステーションから低価格のゲームステーションに至るまで入手可能な3次元グラフィックスハードウェアのアクセラレータが急激に増大している。しかし、対話式のコンピュータグラフィックスは、本当の印象を仮想世界に与えるようなリアリズムのレベルには未だ到達していない。例えば、実時間ゲームにおける一般的な前景部キャラクターは、角張ったシルエット等の不快な人工産物をしばしば出現させている極端にミニマリズムに走ったポリゴンモデルになっている。

【0003】暗示的な面や細別する面等の色々な精巧なモデル化の技術で、増大しつつある複雑な形状の3次元グラフィックスモデルを創造できるようにしている。しかし、より高い等級のモデル化のプリミティブは、結局は、グラフィックスサブシステムによって表現される以前に三角形に分解されている。プリミティブをレンダリングする三角形は、像描写力とコンピュータ負荷との間で適正なバランスをとる上で満足のできるものである。現実味をだすために、生物に見えるモデルは、更により多くの三角形を有し、又は1999年3月6日のエコノミストの73ページ〜74ページの「Smooth Operator :滑らかなオペレータ」でスミス氏が述べているような高度に複雑な形状を必要としており、「リアリティを出すには8000万個のポリゴンを使う」と言っている。

【0004】多数の小三角形を処理することは、帯域幅の隘路となり、過度の浮動小数点数計算を招き、過度のラスタライズ化要件を必要とすることになる。オブジェクトの可視複雑性を高めるために、テクスチャマッピングが導入されてきた。テクスチャは、ポリゴン内部をより詳細に現し、それによってもっと大きくて数の少ない三角形が使われるようにしている。今日のグラフィックスエンジンは、高いテクスチャマッピングの性能を求めて高度に仕組まれている。しかし、テクスチャ写像は、ポリゴンモデルの潜在的な幾何学形状に追従して、平坦な又は若干湾曲した表面上で最も良好な仕上がりにならなければならない。リアルなまたは「生物」の表面は、しばしば、ラスタライズ中に多くの工程で適用されなければならない多数のテクスチャを必要とする。

【0005】フォーンシェーディング（: Phong shading）や、バンプマッピング（bump mapping）や変位マッピング等の進んだレンダリング技術は、大部分の現在の消費者向けのグラフィックスシステムでは取り扱われていない。煙や火や水等のグラフィックス現象は、三角形を使ってテクスチャを出すように表現するのは困難である。

【0006】グラフィックス表現

コンピュータグラフィックスでは、色々なプリミティブグラフィックエレメントを使って多くの異なった方法で3次元空間でオブジェクトを表示することができる。グラフィックオブジェクトを表示するのに一般的に使用されている公知の表示は、暗示的で、幾何学的で、ボリュメトリックであり、点サンプリングに依るものである。

【0007】陰関数表示の表現

陰関数表示の表現において、グラフィックオブジェクトは任意の数学的及び/若しくは物理学的関数から発生される。例えば、中空な球体の輪郭を描くために、人は単に $x^2 + y^2 + z^2 = r$ の関数（デカルト座標における）を表現エンジンに与えればよく、中実球体に対しては、関数は、 $x^2 + y^2 + z^2 \leq r$ である。色や他の材料特性

は、同様に合成して発生されることになる。関数は、色々な幾何学的形状や、物理的物体や現実の又は空想上のモデルを説明するのに使用されることになる。陰関数は、例えば人の形態等の複雑なオブジェクトを合成するには適していない。

【0008】幾何学的表現

伝統的に、3次元オブジェクトは、ポリゴン小面のメッシュとして幾何学的にモデル化されてきた。通常、ポリゴンは三角形と成っている。各小面の寸法は、殆ど小面の領域におけるオブジェクトの湾曲度合に対応するように決められている。多くのポリゴンは、オブジェクトが大きな曲率を有している場合に必要とされるが、比較的平坦な領域に対しては少なくできる。ポリゴンモデルは、バーチャルトレーニングの状況や3次元のモデル化のツールや、ビデオゲーム等の多くの用途に使用されている。特色として、幾何学的表示は、ただ単に、グラフィックオブジェクトの表面輪郭を扱うに過ぎない。

【0009】しかし、小面の寸法が変形されたオブジェクトにおける局所的な曲率の度合には対応できず、結局は三角形が平坦の面を有しているため、ポリゴンモデルが変形されると問題が生じる。更に、変形は、局部領域の相対的解像度を変えることになる。いずれの場合でも、変形された湾曲に従ってオブジェクトを再度調整する必要がある。再調整（ポリゴン化）は、演算時間の観点から比較的費用がかかるために、それは通常事前処理工程として行われる。結果的に、ポリゴンモデルは、動的に変形される必要があるオブジェクトにはうまく合っていない。

【0010】ボリュメトリック表現

代わりの表現では、オブジェクトは、例えばMRIやCT走査等でボリュメトリックデータ集合を生成するために3次元空間でサンプリングされる。各サンプルは、ボクセルと呼ばれている。一般的なデータ集合は、100万のボクセルを有している。ボリュメトリックデータ集合を表現するために、オブジェクトは、一般にセグメント化される。等一表面（Iso-surface）は、特定の容積領域に焦点を合わせるために識別される。例えば、人の頭のボリュメトリックデータ集合は、骨や柔軟組織等の材料特性に応じてボクセルを区分しているであろう。

【0011】ボクセルが多数なために、物質に基づいたモデル化や、ボリュメトリックデータ集合の変形は、依然として非常に演算に対して費用のかかる作業である。しばしば、人はただ表面の輪郭に興味を持ったり、またオブジェクトの内部については事実上無視されることがある。

【0012】点サンプル表現

オブジェクトについての点のサンプル表現は、しばしば、例えば風洞シミュレーションにおける流体の流れをモデル化するのに使用される。方向速度等の或る属性は、流体の流れに渡って個々の点サンプルを追跡した

り、又は全体の流れをビジュアル化するために点サンプルに与えられる。

【0013】点サンプル表現のもう一つ別の適用は、煙やダストや霧等の『雲状』オブジェクトのビジュアル化にある。シェーディングモデルは、雲状オブジェクトを表現するために光を放出する点サンプルに適用される。更に点サンプルは、表面をモデル化するためにエネルギー関数の助けを得てサブ空間に抑制されることになる。点サンプルから成る雲の長所は、雲が非常に変形しやすい事である。短所としては、雲における点サンプルは、分離していて、力を受けると個別に行動することである。更に、従来技術の点サンプルは、ソリッドオブジェクトやモデルの表面を表現するには全く適していない。

【0014】レンダリング上の考慮点

これら従来のプリミティブに対するレンダリング時間は、モデル化されるオブジェクトの複雑さに依存している。例えば、複雑なオブジェクトの幾何学的表示に関しては、ポリゴンは一般的に寸法が非常に小さく、ピクセルの数は非常に少ない程度となっており、またオブジェクトは多くのポリゴンによって表示される。ポリゴンは、通常、三角形を定義している頂点で表示される。

【0015】ポリゴンをレンダリングするために、三角形の投影は、投影に入る各ピクセルの明暗度を算定するために走査一変換（ラスタ化）される。これは、ほんの数個のピクセルが各ポリゴンによって変換される時には、比較的時間のかかる作業と成る。ポリゴンを点サンプルに置き換えて点サンプルを画像へと投影することは、オブジェクトを表現する上でより効率的な技法となる。

【0016】ボリュームをレンダリングするには、幾つかの技法が知られている。一般に、ボリュームレンダリングは極めて複雑である。ボクセルの数が制限されていなければ、実時間レンダリングは、時間が掛かったり、又は実時間の応用には実用的でない。

【0017】離散的パーティクル

1998年7月14日にボノマロフ氏等に付与された米国特許第5,781,194号の『ボクセルベースのオブジェクトの実時間投影』に説明されている実時間レンダリングシステムは、表面のボクセル間に増分ベクトルを使用して表面のボクセル鎖を構築するものである。その投影は、オブジェクトのモデル化と表示に成功しており、非常に詳細な表面領域を示している。硬質体の動きのモデル化は、物質に基づいた方法が使用されていないのでリアリズムに欠けたスクリッピング（scripting）機構の助けを得て実施される。

【0018】レンダリングプリミティブとしての点の使用は、コンピュータグラフィックスに長い歴史を有している。キャットマル氏は、1974年12月のユタ大学の哲学博士論文の『湾曲表面のコンピュータ表示の細別化アルゴリズム』において、幾何学的な細別が究極的に

は表面上の点に到達することを認めていた。パーティクルは、引き続き、他の方法では表現できなかった雲、爆発、火等のオブジェクトに使用された。このことについては、リープス氏の1983年7月のSIGグラフ集録第359～376ページの『パーティクルシステム』におけるファジーなオブジェクトの部類のモデル化技法を参照して下さい。

【0019】ビジュアル上複雑なオブジェクトは、動的に発生される画像スプライトによって表示されて来た。スプライト(sprite)は、描くのが早く、またオブジェクトのビジュアル特徴を十分に保っており、これについては、1998年7月のSIGグラフ集録第231～242ページのシェイド氏等の『層状深度画像: Layered Depth Images』を参照して下さい。同様なアプローチは、高いばば一定のフレーム速度を維持するためにタリスマンレンダリングシステムに使用されていたが、これについては、1996年8月のSIGグラフ集録第353～364ページのトルボルグ氏等の『タリスマン: PC用日用品実時間3次元グラフィックス』を参照して下さい。しかし、平らなポリゴン上へオブジェクトをマッピングすることは、視覚エラーを起こすことになり、視差と非閉塞の作用を許容しない。これらの問題に対処するために、幾つかの方法でピクセル当りの深度情報を、二、三の名称をあげると、層状インボスターや深さを有したスプライトや層状深度画像と色々と呼ばれている画像に加えるようにしている。更に、これらの技法はいずれも、任意の視点から照射され表現される完全なオブジェクトのモデルを与えてはくれない。

【0020】全てのこれらの方法は、オブジェクトや情景を表示するために、視線依存の画像芯出されたサンプルを利用している。しかし、視線依存のサンプルは、オブジェクトの動きや、物性の変化や、光源の位置及び強さの変化を伴った動的情景に対しては効果的で無い。

【0021】1985年のノースカロライナ大学の技術レポート第85-022号の『表示プリミティブとしての点の利用』においてレボイ氏等は、オブジェクトを点表示に変換するプロセスを説明している。そこでは、各ポイントは、位置と色とを有している。彼等は、更に、滑らかな表面として点を表現するプロセスも説明している。点は、0次元のサンプルとしてモデル化されており、オブジェクトオーダの投影を利用して表現されている。レンダリングする時には、多数の点と同じピクセルへと投影され、これらの点の明暗度は、考慮の下でピクセルの最終明暗度を得るべくフィルタにかけられる必要があるであろう。このフィルタリングは、画像における投影された点位置から対応したピクセル中心までの距離に比例して明暗度に重み付けして実施されるが、ところが重みは、表面によるピクセルの部分的被度(partial coverage)に従って正規化されている。被度は、画像空間における投影点の密度を算定して推定され、また重み

は、ガウスのフィルタでモデル化されている。高められた深度バッファ(z-バッファ)は、深度値の小さな領域での点の混和を可能にする許容値と深度比較ができるようにしている。それらの点表示は、いずれの視点からもオブジェクトを表現できるようにしている。

【0022】もう一つ別の技法では、1998年7月のユーログラフィックスワークショップ19998年の集録のレンダリング技法1998年の第181～192ページの『点サンプルレンダリング』においてグロスマン氏等によって説明されているように、点サンプルは、正三角形格子上のオブジェクトの正射投影をサンプリングすることで得られる。正三角形格子は、隣接したサンプリング点間の間隔がより規則正しいので四辺形格子よりも好ましいものであった。

【0023】1996年5月のマサチューセッツ工科大学の技術報告書AIM-1604の『デルタツリー: 画像をベースにしたレンダリングへのオブジェクトを中心にしたアプローチ』においてダリ氏等は、画像ベースのレンダリングへのオブジェクトを中心としたアプローチとしてデルタツリーを導入した。しかし、彼等の方法における視点の動きは、特定の場所に依然として制限されている。

【0024】1999年8月のSIGグラフ集録の第291～298ページの『LDIツリー: 画像ベースのレンダリングのための階層的表現』においてチャン氏等は、画像ベースのレンダリングのために階層的空間区分データ構造にLDIツリーを与えた。

【0025】

【発明が解決しようとする課題】全ての公知の表現には、幾つかの制約がある。従って、必要とされるものは、各々の最良の特長を組み合わせ且つレンダリングを簡略化するオブジェクト表現法である。

【0026】

【課題を解決するための手段】本発明は、鮮やかで変化に飛んだ形状とテクスチャとを有したオブジェクトをインタラクティブなフレーム速度で表現する方法を提供する。本方法は、レンダリングプリミティブとしての表面要素(surfaces、以下サーフェルと称す)をベースとしている。サーフェルとは、グラフィックスモデルの点サンプルのことである。事前処理段階では、複雑な幾何学的モデルの表面は、3つの正射投影視線に沿ってサンプリングが行われる。本発明は、画像空間解像度を利用してオブジェクトを適応できるようにサンプリングを行う。同時に、テクスチャ、バンプ又は変位のマッピング等のコンピュータ使用頻度の高い計算が行われる。ラスター化とテクスチャリングを核となるレンダリングパイプラインから事前処理工程に移すことで、表現コストは、劇的に削減される。

【0027】表現上の観点から、本発明に係るサーフェルレンダリングは、外形の識別を行い、それで、オブジ

ェクト表現をレンダリングに必要とされる必須物に削減する。対照してみると、三角形のプリミティブは、頂点の結合価や隣接性のようなレンダリングには必ずしも利用されなかったり、又は必要とされないデータ等の接続性情報を陰関数表示で記憶している。

【0028】法線や、事前にフィルタに通されたテクスチャや、他のサーフェル毎のデータを記憶することで、高品質のレンダリングプロセスを構築できるようにしている。シェーディングと変換は、フォーン照明やバンブや変位のマッピング及び他の進んだレンダリング特長を達成するために、サーフェル毎をベースにして適用される。

【0029】レンダリングは、更に、インタラクティブなフレーム速度で作動する線よりも色彩を強調するサーフェルレンダリングプロセスを環境マッピングを提供する。階層的フォワード投影アルゴリズムは、スピード品質のトレードオフをつけるために出力ピクセル当りのサーフェル密度を推定できるようにするものである。

【0030】サーフェルレンダリングパイプラインは、従来のグラフィックスパイプラインを補完する。本パイプラインは、レンダリング性能と品質のためのメモリオーバーヘッドをトレードする。本発明は、インタラクティブな3次元適用に、特に非常に詳細な表面を有した生物オブジェクトや、事前処理が論点となっていない適用に適している。これらの特質は、本発明をインタラクティブなゲームにとって理想的なものにしている。

【0031】本発明に係るサーフェルは、インタラクティブなフレーム速度で効率的に複雑な幾何学的オブジェクトを表現する上で強力な範例である。従来のような表面の識別、即ち三角形や四辺形の網目状組織とは違って、サーフェルは、明確な接続性の無い点のプリミティブと成っている。サーフェルの属性は、深度、テクスチャ色、法線等から構成されている。事前処理として、幾何学的オブジェクトのオクトリベースのサーフェルレンダリングが実現される。サンプリング中に、サーフェルの位置と法線は、任意に摂動し、またテクスチャ色の異なったレベルが事前にフィルタに通され、視線に独立した状態でサーフェル毎に記憶される。

【0032】レンダリング中に、階層的フォワードワーピングアルゴリズムは、サーフェルをzバッファ（深度バッファ）に投影する。可視部のスブラッチングと呼ばれている新規な方法は、zバッファにおける可視サーフェルと孔とを決定する。可視サーフェルは、テクスチャのフィルタリングと、フォーン照明と、サーフェル毎の法線を使用する環境マッピングとを使ってシェーディングされる。スーパーサンプリングを含んだ画像再構築の幾つかの方法は、フレキシブルな速度品質のトレードオフを与える。操作が簡単なので、サーフェルレンダリングパイプラインは、ハードウェアの実装に順応できる。サーフェルオブジェクトは、複雑な形状と、低

いレンダリングコストと高い画像品質を与えてくれるものであり、それでそれらをゲーム等の低価格で実時間のグラフィックスに特に適したものとしている。

【0033】より具体的には、レンダリングシステムは、オブジェクトの表面の形状及び陰の属性を記憶するメモリを有している。それら属性は、メモリにおいてオクトリとして配列されている。オクトリは、複数のノードを複数のレベルで配列しており、各ノードは、複数の0次元のn個のエレメントから成る集合を記憶しており、n個のエレメント集合は、グラフィックオブジェクトの表面の一部分の形状及び陰の属性を局部的に近似させており、またn個のエレメント集合は、画像空間のサンプリング解像度を有している。複数の平行な処理パイプラインは、メモリに接続されている。パイプラインは、オクトリのノードのn個のエレメント集合を最低解像度レベルから最高解像度レベルまでトラバースさせることで、オクトリの形状及び陰の属性を選択された向きを有した画像面に投影する。

【0034】グラフィックオブジェクトは、オブジェクトを通して光線を投射することで、サンプリングされる。光線は、オブジェクトを取り囲んだ直交面から発せられる。オブジェクトの表面は、光線が表面と交差する点での形状及び陰の属性を求めてサンプリングされる。各サンプリングされる点のサンプリングされる形状及び陰の属性は、メモリに記憶されたオクトリに記憶される。

【0035】グラフィックオブジェクトの表面点の陰の属性は、各表面点の位置における正接円盤を構成することでフィルタリングされる。正接円盤は、徐々に大きくなる半径を有している。各正接円盤は、テクスチャ空間における楕円に投影される。視野独立フィルタ機能は、表面点に対するテクスチャミップマップを発生するために、各表面点の位置で適用される。フィルタ機能は、投影された正接円盤に等しい範囲を有している。表面点は、深度バッファにおけるピクセルに投影され、また視野依存フィルタ機能は、ピクセルに対する色を決めるために画像バッファにおける各ピクセルに適用される。

【0036】

【発明の実施の形態】我々は、サーフェルとして、即ち「surfels」として表示されるグラフィックオブジェクトについて、画像ベースの適応性のあるサンプリングとオブジェクトベースのレンダリングとについて説明する。図1に示されているように、我々は、グラフィックオブジェクト103の表面102の一部分101を局部的に近似させる陰（shade）及び形状（shape）の属性を備えた0次元のn個エレメント集合体としてサーフェル100を定義している。形状属性は、オブジェクトの位置と向きと深度の情報を含むことができる。陰属性は、テクスチャと材料特性と不透明度とを有することが

できる。以下により詳細に説明するように、我々は、サーフェルを縮小された層状深度立方体（layered depth cube: LDC）のツリーとして記憶する。

【0037】我々の発明のサーフェルレンダリングは、オブジェクト空間への画像空間の投影であり、結果的に任意の2次元のマニホールドと成っている。換言すれば、我々のサーフェル位置属性は、画像空間の解像度を備えたオブジェクト空間の座標を有している。サーフェルのマニホールドは、より複雑な2次元マニホールドを形成するために互いに接続される。マニホールドは、現実的な又は仮想上の任意のオブジェクトを概略示している。

【0038】我々の技術と対照的に、従来技術のレンダリングプリミティブは、オブジェクト空間の解像度によって通常サンプリングされる。我々の表示は、オブジェクトサーフェルと画像面のピクセルとの間のマッピングを定義することで、オブジェクト空間レンダリングと画像空間サンプリングとを組み合わせるようにしている。サーフェルは、画像解像度に応じて発生される。かくして、ピクセルよりもより細かな詳細部は、オブジェクトをサンプリングする際には何ら考慮されていない。オブジェクト空間座標と画像空間解像サンプリングを組み合わせることで、我々は簡単に効率的で高速のレンダリング*

表A

特性	ポリゴン	ボクセル	点	サーフェル
外形	YES	NO	NO	NO
サンプリング	オブジェクト	オブジェクト	オブジェクト	スクリーン
格子	NO	YES	NO	YES
接続性	YES	NO	NO	YES
変形	中程度に困難	困難	容易	容易

【0042】幾つかの方法では、ポリゴンがほぼ1つのピクセルの寸法を有している時、サーフェルは、変換されたポリゴンのピクセルの属性を有している。サーフェルは、更に、抽出された8-接続の表面ボクセルとしても考えられ、そこでは、サーフェルが配置されているセルが1×1×1ピクセルの寸法を有すると共に、6つの隣接サーフェルを有している。サーフェルオブジェクトは、更に、画像格子の解像度に基づいて決められるパーティクルのマッピングとしても考えられ得るものである。

【0043】サーフェルも相違点を有している。例えば、サーフェルは、それらの外形ではボクセルやパーティクルとは違っている。サーフェルは、格子に関してはポリゴンやパーティクルとは違っている。サーフェルは、隣接したエレメントの関連の仕方ではピクセルやパーティクルとは違っている。サーフェルは、それらが予想された出力スクリーンの解像度に従って、またオブジェクト空間の基準に依らずにサンプリングされる事では点とは違っている。サーフェルは、更に、それらが離散的なサーフェルサンプリング格子から生じる明示的ではなく、暗示的な接続性を有している事で点とは違ってい

＊グを行う。我々はピクセルに対するオブジェクトオーダの投影プロセスを説明する。可視性スプラッチング（visibility splatting）と呼ばれている技法を使用して、閉塞されたサーフェルを放棄し、また連続した2次元画像を補間技法を使って再構築する。

【0039】画像空間の解像度に従ったサンプリングは、サンプリングされるオブジェクト空間と画像空間との間に直接的な対応を与える。このようにサーフェルを定義することで、オブジェクトのレンダリングは、オブジェクトの再サンプリングが視野方向のいかにかわらず、レンダリング中に必要とされないと言う点でより容易になる。かくして、「サーフェル化された」オブジェクトのレンダリングすることは、より効率的である。画像空間解像度を備えたサーフェル格子は、我々にいずれの視野方向に対しても事前にサンプリングされたオブジェクトをレンダリングできるようにしてくれる。

【0040】表Aは、従来技術のポリゴン、ボクセル、及び点サンプリングを我々の発明に係るサーフェルと比較している。その表は、我々のサーフェルが公知の従来技術の表示プリミティブと類似した属性を有していることを示している。

【0041】

【表1】

る。

【0044】従来技術のプリミティブと比較すると、我々が表面要素を定義する仕方における最も重要な相違は、サーフェルが画像空間解像度に従ってサンプリングされることである。ピクセルとパーティクルは、通常オブジェクト空間の解像度に従ってサンプリングされる。ポリゴンは、画像解像度でサンプリングされるが、しかし、サンプリングは、サンプリングが視線依存しているためにオブジェクトが変形される時は、投影やレンダリングの直前に行われなければならない。サーフェルを求めるために、画像解像度に対しサンプリングすることが、サンプリングが視線に独立しているので事前処理工程で一度実施される。

【0045】我々の発明に係る画像空間解像度サンプリングでは、グラフィックオブジェクトは、画像再構築が後に続く画像面へのサーフェルの簡単な投影によってオブジェクトの表面を再構築するのに丁度十分なサーフェルを有している。例えば、100×100のサーフェルから成る矩形のサーフェルポリゴンは、画像面上に100×100のピクセルを発生させる。画像面は、画像バッファ内でピクセルとして物理的に表現される。正規

には、画像に対するサーフェルの寄与は、約1対1となる。

【0046】事前処理とレンダリング

我々の発明は、事前処理とレンダリングの2段階で、グラフィックオブジェクトを取り扱っている。事前処理段階では、我々は、グラフィックオブジェクトをサンプリングし、次いでサンプリングされたデータをフィルタに通す。我々は、特定のグラフィックオブジェクトをただ一度だけ事前処理する。サンプリングは、ソフトウェアプログラムによって実施される。これは、一回限りの作業なので、精巧な技法が、できるだけ多くの属性情報をオブジェクトから抽出し、また良質な画像を発生させるためにどの視野方向に対しても表現上効率的なデータ構造へとサンプリングされたオブジェクトを縮小するのに使用される。レンダリング段階では、我々は、データ構造を表現する。ここでは、我々はハードウェアのパイプラインを利用する。実践的に、我々が一度困難な作業を行うと、それで我々は何度も行わなければならない作業は容易になることを体験している。このことは、我々のパイプライン処理されるサーフェルレンダリングをアニメーションの用途に良く適したものとする。

【0047】サンプリングと事前フィルタリング

図2は、サンプリングの事前処理段階200の高レベルのブロック線図である。適応サンプリングプロセス210は、グラフィックオブジェクト201及びそのテクスチャの属性をサーフェル211に変換する。サンプリング中に、我々はサーフェルを3つの直交した層状深度画像(LDI s)に配列するためにレイキャスティング(ray casting)を利用する。LDI sは、各光線—表面の交差点に一つずつ各光線に沿って多数のサーフェルを記憶する。我々は、3つの直交LDI sのこの配列を層状深度立方体(LDC)、すなわち「ブロック」と呼んでいる。例えば、我々は480²の予想された出力解像度に対して512²のサンプリング解像度を使用するようにしている。即ち、我々は、所定の画像品質を提供するためにサンプリング解像度を選択する。

【0048】事前フィルタリング工程220について、以下により詳細に説明する。この工程の主目的は、ブロックの視線に独立したテクスチャ属性を抽出することで*

表B

データフィールド	記憶装置
3つのサーフェルテクスチャミップマップレベル	3×24ビット
法線表に対するインデックス	16ビット
LDI深度値	32ビット
物質表に対するインデックス	16ビット
サーフェル当たりの総バイト数	17バイト

【0052】LDCツリーの寸法は、オクトリのデータ構造では、例えばポインター等のオーバーヘッドのため、サンプリングされたデータよりも約2倍程大きくなっている。LDCツリーは、ランレングス符号化(run length coding)やウェーブレットベースの圧縮(wavelet-bas

*ある。我々のデータ構造では、LDCの「ブロック」は、オクトリ221の各ノードに取り付けられている。オクトリは、コンピュータグラフィックスでは公知であり、例えば1988年1月のグラフィックスに関するACM会報の第7巻第1号61～75ページのヴィーンストラ氏等の「オクトリで表示されたオブジェクトの線図」を参照して下さい。オクトリは、3次元を指し示すために使用されている。我々のLDCツリーの各レベルは、サーフェルオブジェクトの異なった解像度に対応している。データ縮小工程230では、我々はオブションで各ブロックツリーを縮小LDCツリー231に縮小する。好ましくは、その縮小は3:1となる。これで、記憶装置のコストを削減し、更に、レンダリング性能を改善する。

【0049】我々のサンプリング方法の重要で新規な面は、サンプリング形状(外形)とサンプリング陰(テクスチャ色)との間の区別である。サーフェルは、表面位置等の形状属性と、例えば図1における表面法線104等の向きを記憶する。我々の好適な実施例では、x-y位置は、LDCツリー221におけるブロック(ノード)の位置によって暗示的に定義される。即ち、陽関数表現のx-y座標は、記憶されていない。深度情報(z座標)は、オクトリに陽関数表現で記憶されている。表面の向きは、図1を見て判るように、表面法線104によって与えられている。法線を実際に記憶する代わりに、我々は、反射と環境マッピングのシェーディングの間に使用される定量化された法線表に対するインデックスを記憶させている。上述のように、形状属性は、オブジェクト空間に基づいている。

【0050】陰(shade)は、多くのレベルの事前にフィルタに通されたテクスチャ色として表現されている。我々は、この新規な階層的な色情報をサーフェルテクスチャミップマップ(surfel texture mipmap)と呼んでいる。事前フィルタリング工程220中に、パンプや変位のマッピング等の他の視線独立方法も、抽出された形状と陰の属性に対して実施される。表Bは、サーフェル当りの最小記憶要件を示している。

【0051】

【表2】

ed compression)の技法によって実質的に圧縮される。

【0053】レンダリングパイプライン

図3は、我々のサーフェルレンダリングパイプライン300を示している。パイプラインは、遠近投影法を使用して画像面399のピクセルにLDCツリーブロックの

ブロック（ノード）を階層的に投影する。レンダリング目的の画像面の向きは任意であり、またサンプリング中に使用される3つの直交した深度画像の向きとは異なっている事に注目されたい。

【0054】レンダリングは、ブロック選択310と高速増分フォワードワーピング（fastincremental forward warping）320によって加速される。我々は、レンダリング速度と画像再構築の品質とを制御するために出力ピクセル当りの投影サーフェル密度を推定する。

【0055】深度バッファ（zバッファ）は、可視性スブラッチング330と呼ばれている新規な方法と共に、可視性の問題を解決する。ここでは、各サーフェルでの正接円盤は、表面の孔を検出するために、且つ隠された（閉塞された）サーフェルが再構築プロセスで使用されるのを防ぐためにzバッファへ走査変換される。

【0056】可視サーフェルのテクスチャ色は、サーフェルのテクスチャミップマップの適切なレベル間の線形補間法を使用してフィルタに通される（340）。各々の可視サーフェルは、例えばフォーン照明や反射マッピングを使ってシェーディングされる（350）。最終工程360は、孔の充填やアンチエイリアシングを含めて、可視サーフェルからの画像再構築を実行する。一般に、出力画像の解像度と深度バッファとは同じである必要はない。事前処理工程200とレンダリング工程300の段階を次により詳細に説明する。

【0057】サンプリング

適応サンプリング210中に、我々の最終目標は、最少の冗長度でオブジェクトの形状と陰の属性の最適なサーフェル表示を見出すことである。大部分の従来技術のサンプリング方法は、曲率やシルエット等のオブジェクト表面の幾何学的パラメータに基づいてオブジェクト識別を行う。そのオブジェクト空間の識別は、一般に、レンダリングのために余りにも多くの又は余りにも少ないプリミティブを導くことになる。我々のサーフェル表示において、オブジェクトサンプリングは、画像空間の解像度に整合されており、即ち、サンプリングは、実質的に画像の予想された出力解像度に適合している。

【0058】LDC-サンプリング

図4（a）に示されているように、3つの層状深度画像（LDI1）401、（LDI2）402の内の2つのみに対して、我々は、立方体の3側面から層状深度立方体（LDC）と呼ばれている3の直交LDIにグラフィックオブジェクト410をサンプリングする。上述のように、サンプリングは、予想された出力画像空間解像度で実施される。レイキャスティングは、裏面を含むオブジェクト表面411との光線の全ての交差点を記録する。LDI1との交差点は、円として図示されており、またLDI2との交差点は、方形として図示されている。図4の（b）は、（a）の一部分の拡大図である。

【0059】各交差点において、我々の処理は、浮動小

数点深度データと他の形状及び陰の属性と共に0次元サーフェル412を発生させる。バンプや変位のマッピングのための表面の又は外形の摂動は、サンプリングの前に、又はプロシージャシェーダを使ったレイキャスティング中に外形に基づいて実施される。LDCサンプリングは、我々に、多くの任意に向けられた深度画像から構築するのは難しいと思われる階層的データ構造を容易に構築できるようにしてくれる。

【0060】適切なサンプリング解像度

もし、LDCピクセル間隔 h を考慮すれば、我々は、三角形の頂点として発生されたサーフェルを使用してオブジェクト表面上にドロネの三角形網を構築することで表面上に結果的に生じるサンプリング密度を決定できる。このサンプリングプロセスによって発生される仮想の三角形メッシュは、 $\sqrt{3}h$ の最大側辺長 S_{max} を有している。最小側辺長 S_{min} は、2本又は3本のサンプリング光線が同じ表面位置で交差する場合にゼロになる。我々が少なくとも一つのサーフェルが正射投影と単位拡大のために各出力ピクセルフィルタの支持体に投影されることを保証すると、オブジェクトは適切にサンプリングされる。オブジェクト空間における隣接サーフェル間の S_{max} が、即ち最大距離がピクセルの再構築フィルタの半径よりも小さい時に、その条件はかなえられる。一般に、我々は、拡大と遠近投影法の効果によってLDC解像度をこれよりも若干高くなるように選択している。例えば、我々が 480^2 の画像解像度に対して 512^2 でサンプリングを行うのを思い出して下さい。

【0061】テクスチャの事前フィルタリング

特長として、我々は、サーフェルのテクスチャを事前にフィルタリングし（220）、またテクスチャは、再処理段階中にオブジェクト空間に写像される。視線依存のテクスチャエイリアシングを防ぐために、我々は、更に、以下に説明するようにレンダリング中にサーフェル毎のテクスチャのフィルタリングを適用する。

【0062】オブジェクト空間に対し、図5（a）に示されているように、我々は、各サーフェル506に円501の中心を合わせる。円の正接面は、テクスチャ空間におけるフィルタの設置面積の範囲を確定する。我々は、これらの円501を正接円盤と呼んでいる。円盤は、対応したサーフェル506の向き502に直交している。

【0063】図5（b）に示されているように、各正接円盤は、表面の予め定義されたテクスチャパラメータ化を利用してテクスチャ空間における楕円503へマッピングされる。ガウシアンカーネルが、テクスチャのフィルタリングに使用される。結果的に生じた色がサーフェルに割り当てられる。適切なテクスチャの再構築を可能にするために、テクスチャ空間における円形や楕円形のフィルタ設置面積（点線）は、図5（a）～（b）に示されているように互いに重ね合っている。

【0064】結果的に、我々は、正接円盤の半径として、オブジェクト空間における隣接サーフェル間の最大距離である $S_{\max} = \sqrt{3}h$ を選択する。このことは、通常、正接円盤がオブジェクト空間において互いにオーバーラップし、且つテクスチャ空間におけるそれらの投影がオーバーラップすることを保証するものである。我々は、可視性を決定するために変形されたz-バッファ充填法を使用するので、下記のように、必ずしも全てのサーフェルが画像再構築に利用できるものではない。このことは、テクスチャエイリアシングの人工物を生むことになる。従って、我々は、サーフェル当たり幾つかの、一般には少なくとも3つの事前フィルタリングされたテクスチャサンプルを記憶させるようにしている。正接円盤は、徐々により大きくなる半径を有している。各々の円盤は、テクスチャ空間にマッピングされ、事前フィルタリングされた色を算出するために使用される。我々は、事前フィルタリングされた色をサーフェルテクスチャミップマップと呼んでいる。図5(b)は、テクスチャ空間における徐々に大きくなっている楕円正接盤の楕円形設置面積503~505を示している。

【0065】データ構造

我々は、サンプリング中に獲得されるLDCを記憶させるために効率的な階層的データ構造を利用する。LDCオクトリ221は、我々にピクセル当りの投影サーフェルの数を推定できるようにすると共に、より高い画像品質に対する表現速度に折合いを付けるようにしてくれる。

【0066】LDCツリー

我々は、最高解像度のLDCのサブサンプリングされるバージョンであるLDCをオクトリの各ノード(ブロック)に記憶させることで画像再構築中の再サンプリングとスブラッチングを回避する。我々のオクトリは、帰納的に底上げから構築される。外形サンプリング中に獲得される一最高解像度のLDCは、LDCツリーの最低レベル($n=0$)に記憶され、また最低解像度は、トップレベルに記憶される。

【0067】2つの寸法に対し、図6(a)~(b)に示されているように、各LDCは、ユーザ指定の寸法601を持ったブロックに細分される。図6(a)は、2次元図を利用したLDCツリーの最高解像度のブロックを示している。オクトリのより高いレベルでの、即ちより低い解像度でのブロック(ノード)は、2部分から成るように、即ち2の累乗倍でそれらの子ノードをサブサンプリングすることで構築される。図6(b)は、LDCツリーのレベル $n=1$ を示している。オクトリのより高いレベルでのサーフェル602は、レベル0のLDCにおけるサーフェル604を参照しており、即ち、階層の幾つかのブロックに現れるサーフェルは、ただ一度だけ記憶され、またブロック間で分担されている。

【0068】もし、最高解像度のLDCがピクセル間隔

h を有していれば、レベル n でのLDCは、ピクセル間隔 $2^n h$ を有している。LDCツリーの高さは、ユーザによって選定される。一つ高さを選定すると、階層を平坦にし、最高解像度のLDCのみを記憶する。LDCツリーは、当然にサーフェルオブジェクトについて詳細レベルの表示を記憶するので、その最低解像度は通常オクトリの高さを確定する。

【0069】図6(a)において白い正方形として示されている空ブロック603は、LDCツリーに記憶されていない。結果的に、ブロック寸法601は、最高解像度のLDCの寸法に関連付けられておらず、任意に選択される。ブロック寸法 $b=1$ を選択することは、LDCツリーを完全な容積測定のおクトリ表示にする。

【0070】3:1の縮小

記憶装置とレンダリング時間を縮小するために、LDCツリーを層状深度画像へブロック毎に基づいて任意に縮小することは有益である。これは、一般にワーピング速度での3倍増に相当しているので、我々は、この工程を3:1への縮小と呼んでいる。先ず第1に、我々は、ターゲットLDIとしてブロック内に一つのLDIを選択する。我々は、ターゲットLDIのピクセルに対して2つの残っているLDIをワーピングし且つ再サンプリングする。

【0071】図7(a)及び(b)に示されているように、図7(a)におけるサーフェル701~712は、図7(b)に示されるようなサンプリング光線交差部703~704の格子箇所へ再サンプリングされている。例えばスブラッチング等のより精巧な事前フィルタリングも実行され得るのであるが、我々は最も近い隣同士の補間法を利用している。再サンプリングされたサーフェルは、縮小されたLDCツリー231に記憶されている。

【0072】縮小と再サンプリングのプロセスは、形状と陰の両方に対してサーフェル表示の品質を低下させる。同じ表面から再サンプリングされたサーフェルは、非常に異なったテクスチャ色と法線705とを有するのであろう。サーフェルが同じ表面に属しているかを確定するしきい値に対して我々は距離を比較できよう。しかし、異なった表面からのサーフェルは、しきい値ぎりぎりになるであろうし、それは、通常薄い構造に対して起きる。これは、オブジェクトの運動中に悪化される色及び明暗度の人工物をひき起すであろう。しかし、実際には、我々は3:1の縮小によって厳しくなった人工物に遭遇したことは無い。我々のレンダリングパイプライン処理は、LDCとLDIを同様に扱うので、我々はLDCとして薄い構造を有したブロックを記憶させることができるが、他方、全ての他のブロックは、単一のLDIに縮小される。

【0073】我々は、3:1の縮小後にオブジェクトの表面上のサーフェル密度についての限界を確定できる。

10

20

30

40

50

ピクセル間隔 h を備えたターゲットLDIを考慮すれば、オブジェクト表面上の隣接サーフェル間の最大距離は、オリジナルのLDCツリーにおけるように、 $S_{max} = \sqrt{3}h$ として表現される。サーフェル間の最小距離は、表面上の仮想のドローネー三角形をより一様にしながら、冗長なサーフェルの除去によって $S_{min} = h$ まで増大する。

【0074】レンダリングパイプライン

レンダリングパイプライン300は、サーフェルのLDCツリー221又は縮小されたLDCツリーを採用し、特定の画像面の向きのためにブロックの階層的可視性選別とフォワードワーピングとを利用してそれを画像399として表現する。階層的表現は、更に、我々に出力ピクセル当りの投影されたサーフェルの数を推定できるようにもしてくれる。最大レンダリング効率のために、我々は、ピクセル当り約一つのサーフェルを投影すると共に出力画像におけるものと同じZバッファ用解像度を使用する。最大画像品質のために、我々は、Zバッファのより細かい解像度と高品質画像の再構築とを使用してピクセル当り多数のサーフェルを投影する。

【0075】ブロック選択

我々は、トップ、即ち最低解像度のノードからボトム又は最高解像度のノードまでLDCツリーをトラバース(traverse)する。各ブロック/ノードに対して、我々は、先ず第1に、ブロック境界ボックス311を使用して視線の錐台形の選択を行う。視野方向は任意なので、異なった視線はオクトリの異なった部分を明らかにするであろう。次に、我々は、ブロックの後面選択と同じ事を実施するために可視性錐体を利用する。サーフェルの法線を使用して、我々は、高速で保守的な可視性テスト(ブロックにおけるサーフェルは、何ら錐体内のいずれの視点からも見ることができない)を提供してくれるブロック当りの可視性錐体を予め演算する。従来技術の点サンプリングレンダリングとは対照的に、我々は、LDCツリーにおいて階層的に全ての可視性テストを実施し、それで我々のテストをより効率的にする。

【0076】ブロックのワーピング

図8に示されているように、投影されるオクトリレベルを選択するために、我々は、各ブロックに対して、ピクセル当りのサーフェルの数を控えめに推定する。我々は、高速レンダリングのためにピクセル当り一つのサーフェルを、又はスーパーサンプリングのためにピクセル当り多数のサーフェルを選択する。ピクセル当りのサーフェルの数は、 d_{max} によって決定される(801)。値 d_{max} は、画像空間における隣接したサーフェル間の最大距離である。

【0077】我々は、ブロック境界ボックス311の4つの主対角線312を投影することでブロック当りの d_{max} を推定する。正射投影のために、それらの最大長さは、 d_{max} についての上部境界である。正射投影を用い

ることで生じるエラーは、ブロックが一般に少数のピクセルに投影するので小さい。

【0078】レンダリング中に、LDCツリーはトップからボトムにかけてトラバースされる。各レベルで、 d_{max} はピクセル再構築フィルタの半径 r 802に比較される。もし、現在のブロックの d_{max} が r よりも大きければ、それでその子ノードは、トラバースされる。我々は、 d_{max} が r よりも小さいブロックを投影し、それでピクセル当り約一つの面エレメントを表現する。ピクセル当りのサーフェル密度は、例えば r をサブピクセルの対辺とする等して、より小さな r を選択することで増大される。フォワードワーピング中に、 d_{max} は、可視性スブラッチングと画像再構築との段階において引き続いて使用するために各投影されたサーフェルと共に記憶される。

【0079】LDCブロックの位置属性を画像空間にワーピングするために、我々は、最適化増分ブロックワーピングを使用する。我々のサーフェルの位置属性は、オブジェクト空間の座標で表現されることを思い出して下さい。従って、我々は、オブジェクトから画像空間にワーブする。このワーピングは、我々のLDCが規則正しいので高効率である。各ブロックにおけるLDIは、独立してワーブされ、それで、我々に幾つかの又は全てのブロックが上述のように3:1の縮小後に単一のLDIに縮小されてしまったLDCツリーを表現できるようにする。

【0080】可視性スブラッチング

遠近法投影や、高いZバッファ解像度や、拡大又はズームは、サンプリング不足か、又はZバッファ(深度バッファ)の『孔』を生むことに成るであろう。Zバッファピクセルは、ピクセルが少なくとも一つの対応した可視サーフェルか、又は背景のピクセル投影を有していない時には、孔として定義される。孔は、画像再構築のためにマークされなければならない。我々は、我々の新規なマーキングアプローチを可視性スブラッチングと呼んでいる。画像再構築について、以下に説明する。

【0081】深度バッファ

我々は、図11の工程950に示されているように、深度値を有したZバッファを用意している。Zバッファの各ピクセルは、最も近いサーフェルに、即ち最小深度(Z)値を有したサーフェルに対するポインターと、現在の最小深度値を保管する。ピクセルは、更に『孔』としてマークされたり、又はされないように成っている。我々のZバッファのピクセルは、例えば『無限大』等の最大深度値や、背景情景で初期化され(951)、また、孔では初期化されない。

【0082】サーフェル深度は、最も近い隣接したものの同士の補間法を使用してZバッファに投影される(952)。サーフェル深度が、各サーフェルと共に記憶されることを表Bから思い出していただきたい。Zバッファ

10

20

30

40

50

は、品質と速度との間に良好なトレードオフを提供してくれ、また我々のZバッファは、OpenGL™（登録商標）等の従来からのポリゴングラフィックスレンダリング方法と統合される。Zバッファピクセルの深度値は、もしサーフェルの深度値（ s_d ）がピクセルの深度値（ p_d ）よりも小さければ、ただ上書きされるだけである（953）。かくして、他の表面特長の前に存在している表面特長のみが可視ことになる。工程945では、正接円盤501は、オブジェクト空間における各サーフェル502に対して構築される。

【0083】図9～図10に示されているように、我々は、孔や後向き表面による可視性の問題を正しく解決するために、サーフェルの正接円盤501の投影をZバッファ900へ走査変換する。構築された正接円盤945は、 $r_t = S_{max} 2^\circ$ の半径を有していて、この場合、 S_{max} はオブジェクト空間における隣接サーフェル間の最大距離であり、 n はブロックのレベルである。円盤は、サーフェルの法線502によって向きを決定している。

【0084】投影954後の図10に示されているように、正接円盤は、サーフェル周りに楕円901を形成する。我々は、部分的に軸整合された境界箱902で楕円901を近似している。境界箱の平行四辺形は走査変換され、また各Zバッファピクセルは、サーフェル法線N502に基づいて適当な深度で充足される。即ち、もし深度値が以前に記憶された深度値よりも小さければ、記憶されていた深度値は上書きされる。

【0085】我々は、我々の可視性スラッシングが計算を簡略化するために工程955で正射投影を利用する。投影された楕円の短軸 a_{n1} 、911の方向は、サーフェル法線Nの投影に平行になっている。長軸 a_{n2} 、912は、 a_{n1} に直交している。長軸の長さは、図8の d_{max} によって近似される s_{max} の投影となっている。この近似は、投影中のLDCツリーの向きと拡大を考慮に入れている。

【0086】次に、我々は、例えば図10におけるy-軸913等の a_{n1} に最も平行になっている座標軸を算定する。境界箱の短辺は、走査変換を簡略化するためにこの座標軸線に整合された線となっている。境界箱の高さ h_{914} は、楕円の座標軸線との交差によって決定される。境界箱の幅 w_{915} は、長軸線と楕円との交差点の頂点をx-軸上に投影することで決定される。

【0087】値 $\partial z / \partial x$ 及び $\partial z / \partial y$ は、画像のx及びyの方向に対するサーフェル深度zの偏導関数である。これらは、正射投影なので一定であり、また単位法線Nから算定される。走査変換中には、境界箱内部での各ピクセルにおける深度が、偏導関数 $\partial z / \partial x$ 及び $\partial z / \partial y$ を使用して算定される。更に、我々は、小さなしきい値 ϵ を各投影されたz値に付加する。しきい値 ϵ は、円盤の下に在るがしかし依然として前景部表面上に

存在しているサーフェルが突然放棄されるのを防止している。工程956では、ピクセル（ p_d ）の深度値は、もし $t_d < p_d$ であれば、投影された正接円盤（ t_d ）の深度値で上書きされる。

【0088】もし、表面が極端に湾曲されていれば、それで、正接円盤は、表面を完全に覆うことがなく、潜在的に引裂と孔を残すことになる。更に、極端な遠近投影は、正射投影を実際の投影正接円盤に対して悪く近似させる。しかし、実際には、我々はこれを大きな問題とは見ていなかった。もし、投影された正接円盤が円形であれば、即ち円盤がほぼ視線方向に平行となっていれば、それで境界箱の平行四辺形は、悪い近似形となっている。この場合、我々は正方形の境界箱を代わりに使用する。

【0089】Zバッファ深度値を決定するための我々の方法も、オブジェクトのボクセルや他の従来の点表示と共にピクセルへラスタ化されるポリゴンと使用されることに注目すべきである。我々の方法は、グラフィックオブジェクトの基礎を成す表示からは独立していずれのZバッファにも存在し得るものである。

【0090】テクスチャの事前フィルタリング

上述のように、LDCツリーにおける各サーフェルは、サーフェルのテクスチャミップマップに、幾つかの事前フィルタリングされたテクスチャ色を記憶させている。表現中に、サーフェル色は、オブジェクトの縮小と表面の向きに応じてサーフェルテクスチャのミップマップ色から線形補間法で決められる。

【0091】図12（a）は、Zバッファに投影されサンプリングされた表面の全ての可視サーフェルを示している。サーフェルの中心周りの楕円1001は、上述のように、最高解像度のテクスチャの事前フィルタリングの占有面の投影を記している。事前フィルタリング中に、我々は、占有面で前表面を覆うようにしていることに注目して下さい。図12（b）では、Zバッファのピクセル当りのサンプル数は、Zバッファ深度テストを適用する事によって1に制限されている。Zバッファにおけるサーフェルのポインターは、もう一つ別のより接近したサーフェルが同じピクセルに配置されるともう一つ別のポインターで代替される。

【0092】テクスチャの占有面による表面の被覆で現れるギャップを満たすために、残っているサーフェルの占有面は拡大されなければならない。もし、サーフェルが或るZバッファピクセルにおいて放棄されれば、それで、我々は、放棄されたピクセル周りの3×3近辺におけるZバッファピクセルが孔ではないと想定することができる。かくして、各サーフェルのテクスチャ占有面が少なくともZバッファの領域を覆うと、ギャップは充填される。結果的に、投影された占有面の楕円は、最悪の場合 $\sqrt{2} s$ の短半径を持たなければならないが、そこで s はZバッファピクセル空間である。我々は最悪のケー

スを見捨てる、サーフェルがZバッファのピクセル中心に投影されることを意味している $\sqrt{2}s/2$ を使用する。図12(b)は、投影されたサーフェル周りの楕円として尺度合わせされたテクスチャ占有面1002を示している。

【0093】図13に示されているように、我々は、適当なサーフェルのテクスチャミップマップのレベルを選定するために視線依存のテクスチャフィルタリングを利用する。半径 $\sqrt{2}s/2$ を有した円1101は、画像空間のピクセルを通して、表面の正接面1102上に視点1103の方向から投影され、正接面に楕円1104を発生する。ピクセルの投影は、正射投影で近似される。等方性のテクスチャマッピングと同様に、投影された正接空間の楕円の長軸がサーフェルのミップマップのレベルを決定するために使用される。サーフェルの色は、最も近い2つのミップマップレベルの間で線形補間法によって決定される。これは、3つの線で囲まれたミップマップにおけるように8個のサンプルの補完とは反対に、2つのサンプル間の線形補間法である。

【0094】シェーディング

従来技術では、照明モデルは一般に可視性テストの前に適用される。しかし、本発明に係る可視性スブラッチング後に据え置かれたシェーディングは、不必要な作業を回避する。更に、従来技術のパーティクルシェーディングは、画像空間への法線の変換を回避するためにオブジェクト空間で通常実施される。しかし、我々は、上述のように、我々の可視性スブラッチング中に法線を画像空間に既に変換している。手近の変換された法線に関連して、我々は、広域的効果でサーフェル毎のフォーン照明モデルを算定するために立体の反射率と環境マップを利用する。サーフェル毎の法線によるシェーディングで、光線追跡品質となっている鏡による光彩の有る場面を生む。

【0095】画像再構築とアンチエイリアシング

投影されたサーフェルから連続した表面を再構築するには、基本的には、分散したデータの補間法上の問題がある。スブラッチング等の従来技術の技法と対照的に、我々は画像再構築から可視性算定を分離している。我々は、上述のように、我々の画期的な可視性スブラッチング中にZバッファピクセルに孔で印を付ける。これらの孔ピクセルは、それらが何ら可視サンプルを含んでいないので画像再構築中には使用されない。

【0096】図14(a)～(b)は、本発明に係るZバッファにおける画像再構築を示している。図14

(a)では、画像(フレーム)バッファが、Zバッファと同じ解像度を有している。サーフェルは、クロスハッチングで示されているように、最も近い隣接したものの同士の補間法を使用することでピクセル中心1201にマッピングされる。孔1202は、黒の×で印が入れられている。

【0097】フォワードワーピング中には、ブロックの隣接した投影サーフェル間の最大距離の推定値としての各サーフェル記憶 d_{max} を思い出していただきたい。この距離は、少なくとも一つのサーフェルを含んでいるピクセルフィルタの最小半径の良好な推定値である。孔を補完するために、我々は、例えば半径が孔ピクセルの中心に位置決めされた d_{max} よりも若干大きな半径を有した半径方向に対称なガウスのフィルタを使用することになる。代わりに、孔を充填するために、我々は1996年8月のSIGGRAPH集録の43～54ページの「Lumigraph: Lumigraph」においてゴートラー氏等によって説明されているように引き押し方法(pull-push method)を適合化することもできる。

【0098】図14(b)に示されているように、高品質の代替案は、スーパーサンプリングを行う。ここでは、出力画像の解像度はZバッファ解像度の半分か、または幾分か他の残りとなっている。スーパーサンプリングの表現は、前述の通りである。画像再構築中には、我々は、サブピクセルの色を事前フィルタリングするために全ての出力ピクセルの中心にガウスのフィルタを設置する。フィルタの半径は、少なくとも一つのサーフェルを覆うためにまた d_{max} となっている。最小半径は $\sqrt{2}s/2$ であり、但しsは出力ピクセルの辺長さである。

【0099】更にもう一つ別の実施例では、我々は、1996年8月のSIGGRAPH集録第477～484ページの「アニメーション用の色彩を強調したレンダリング」においてマイアー氏によって説明されているように色彩を強調したレンダリングアルゴリズムのインタラクティブなバージョンを適合させている。我々の適合化では、我々は、Zバッファにおける各可視サーフェルにおけるピクセル毎のアルファバージョンで、向きを取ったブラシテクスチャを表現する。テクスチャは各可視サーフェルに中心が取られており、そのRGBAピクセルは、サーフェル色が掛け合わされる。「ブラシ」の向きを取るために、画像空間におけるサーフェル法線は、画像面に正射投影され、テクスチャは、結果的に生じるベクトルと軸整合されている。テクスチャは、それで、テクスチャスブラッチングと同様に画像空間のラスタ化を使用して出力ピクセルにマッピングされる。各サーフェルのブラシ寸法は、同じになるか、又はサーフェル毎の法線又はテクスチャの派生物は、テクスチャに尺度を当てるのに使用される。代わりに、各サーフェルは、ブラシ型式と向きと寸法と共に係数を表に記憶できるであろう。マイアー氏とは対照的に、我々は可視性スブラッチング後にテクスチャスブラッチングを行う。

【0100】どのようにして出力ピクセルの色が正規の表現や、孔の無い場合でのスーパーサンプリングに対して決定されるのかを説明するのは有益である。正規の表現のためには、ピクセル色は、Zバッファにおける最も

接近したサーフェルから最も近い隣接したものの補充によって決定される。そのサーフェルの色は、2つのサーフェルのテクスチャミップマップレベル間で線形補間法によって決定される。かくして、出力ピクセルの色は、2つの事前フィルタリングされたテクスチャサンプルから決定される。スーパーサンプリングの場合には、一つの出力ピクセルは、zバッファのサブピクセル当り一つのサーフェルのフィルタリングされた色を含んでいる。かくして、8個までの事前フィルタリングされたテクスチャサンプルは、2×2のスーパーサンプリングのための出力ピクセルに寄与しよう。これで、3本の線に囲まれたミップマップと同様の画像品質を発生する。

【0101】階層的密度推定、可視性スブラッチング、及びサーフェルミップマップテクスチャフィルタリングを行う我々の方法は、比較可能な従来技術のレンダリングシステムよりもよりフレキシブルな速度-品質のトレードオフを与えてくれる。

【0102】

【発明の効果】我々のサーフェルレンダリングの主要な長所は、どんな種類の合成オブジェクトも、又は走査されたオブジェクトもサーフェルに変換されるという事である。例えば、我々は、非合成のオブジェクトの容積データ、点の雲、及びLDIをサンプリングできる。LDCツリーの閉塞の両立可能なトラバースを利用して、我々はオーダに独立した透明度と真正な容積の表現をできるようにしている。サーフェルレンダリングパイプラインのハードウェア設計は、真っ直なものである。ブロックワーピングは、zバッファのテスト用にただ2つの条件命題を含んでいる。我々は、クリッピング計算を実施する必要がない。可視性スブラッチングと画像再構築等の全てのフレームのバッファ操作は、標準的なラスタ化とフレームバッファ技法を使用して実行される。我々のレンダリングパイプラインは、テクスチャマップからのテクスチャの自動照合等の逆関数計算を何ら使用していない。実行時間のテクスチャフィルタリングは、我々のパイプラインで簡単になる。形状と陰の情報は、サーフェルの位置データと同時にパイプラインに装荷されるので、高度なデータの所在特定が存在している。結果的に、データをキャッシュに入れることは、更に性能を改善することになる。

【0103】我々のサーフェルの表現は、非常に高度に複雑な形状と陰を有した生物モデルに対して理想的である。我々は、パイプラインではなく、演算段階でラスタ化とテクスチャフィルタリングを行うので、ピクセル当りの表現コストは、劇的に削減される。レンダリング性能は、ワーピングとシェーディングと、画像再構築とによって基本的に決定される。これらの作業は、容易にベクトル化と、並列法と、パイプライン処理とを活用でき

る。我々のサーフェルレンダリングパイプラインは、幾つかの速度-品質の折合いを行う。画像再構築とテクスチャフィルタリングを分離することで、我々は、比較可能な従来技術の点サンプリングアプローチよりもより高度な画像品質を達成する。我々は、孔の検出に非常に効果があって且つ画像再構築性能を高める可視性スブラッチングを導入している。スーパーサンプリングによるアンチエイリアシングは、当然我々のシステムに統合されている。我々のパイプラインは、インタラクティブなフレーム速度で高度な画像品質を得ることができる。

【0104】本発明を好適な実施例によって説明してきたが、色々な他の適合化の改造も本発明の精神と技術的範囲内で行われることは理解すべきである。従って、本発明の真正な精神と技術的範囲に入るような全ての变形と改造をも含めるのが従属請求項の目的である。

【図面の簡単な説明】

【図1】 本発明に係るグラフィックオブジェクトのサーフェルの線図である。

【図2】 事前処理のサンプリング段階のブロック線図である。

【図3】 サーフェルレンダリングパイプラインのブロック線図である。

【図4】 層状深度立方サンプリング方法の線図である。

【図5】 テクスチャの事前フィルタリングの線図である。

【図6】 LDCツリーの2つのレベルの線図である。

【図7】 LDC削減の線図である。

【図8】 サーフェル密度の推定の線図である。

【図9】 可視性スブラッチングの線図である。

【図10】 可視性スブラッチングの線図である。

【図11】 zバッファにおける深度値の記憶方法のフロー線図である。

【図12】 投影されたサーフェルのテクスチャミップマップの線図である。

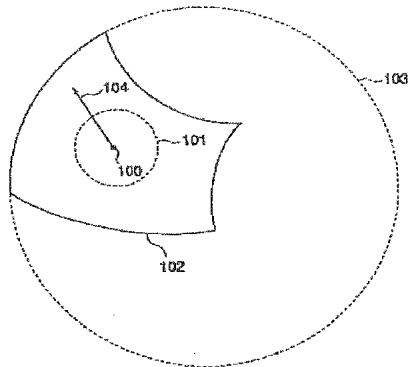
【図13】 視線依存のテクスチャフィルタリングの線図である。

【図14】 画像再構築の線図である。

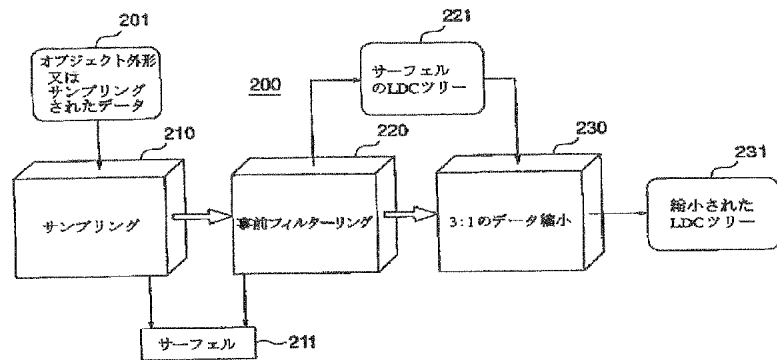
【符号の説明】

201 オブジェクトの外形又はサンプリングされたデータ、210 サンプリングプロセス、211 サーフェル、220 事前フィルタリング工程、221 サーフェルのLDCツリー、230 データ縮小工程、310 ブロック選択、320 フォワードワーピング、330 可視性スブラッチング、340 テクスチャのフィルタリング、350 繰り延べられたシェーディング、360 画像再構築とアンチエイリアシング。

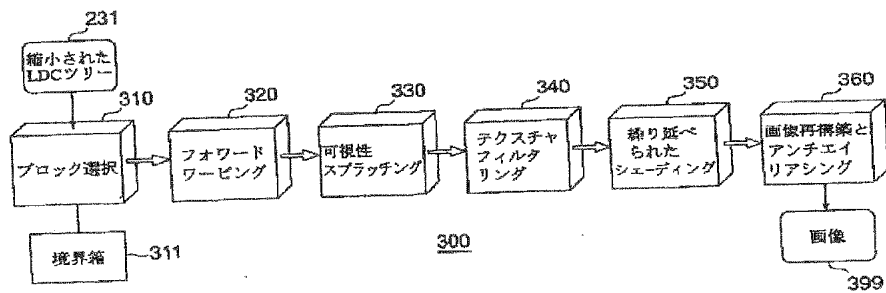
【図1】



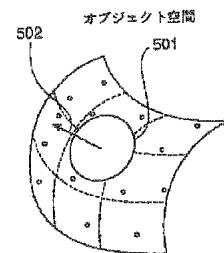
【図2】



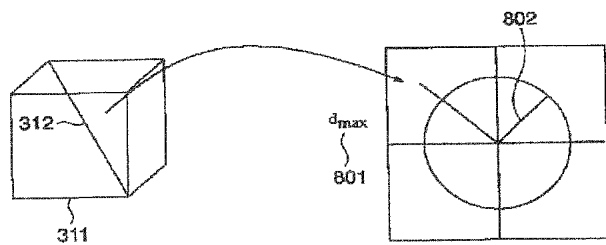
【図3】



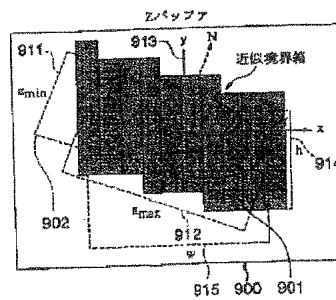
【図9】



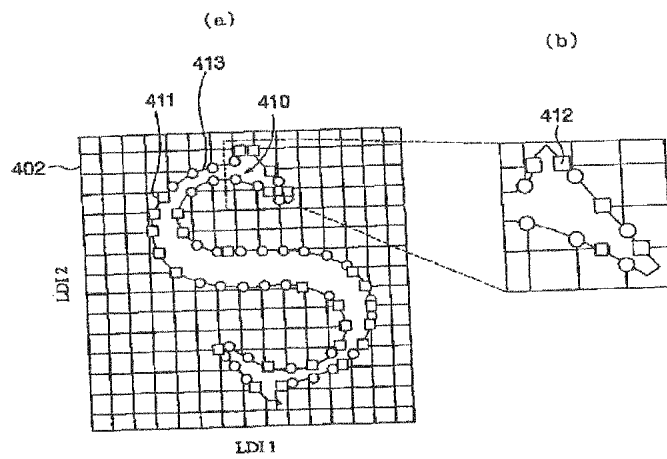
【図8】



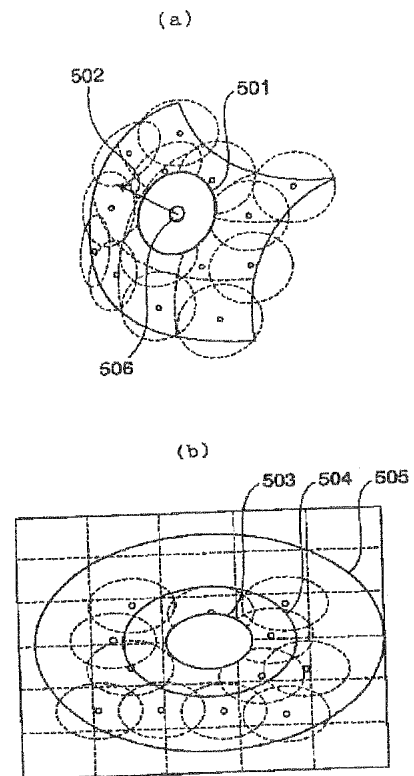
【図10】



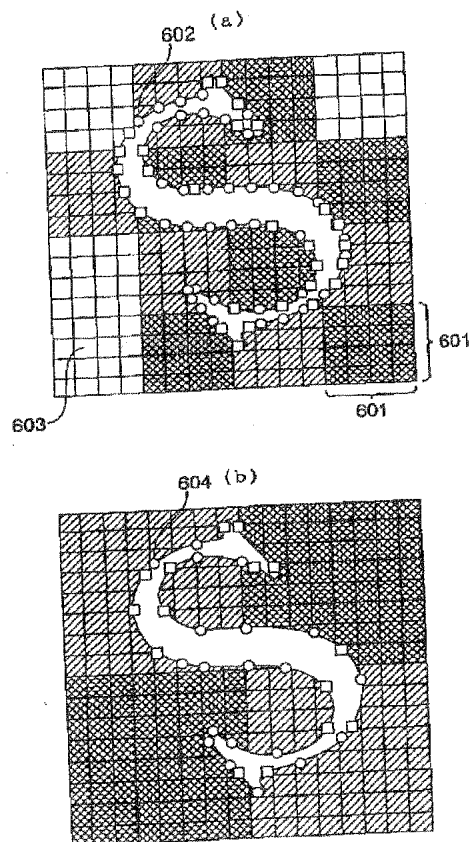
【図4】



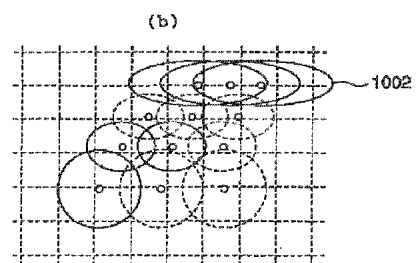
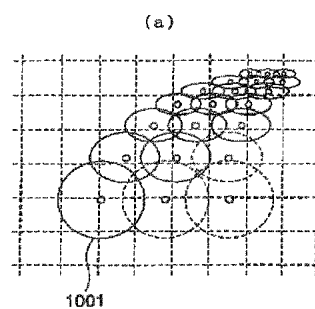
【図5】



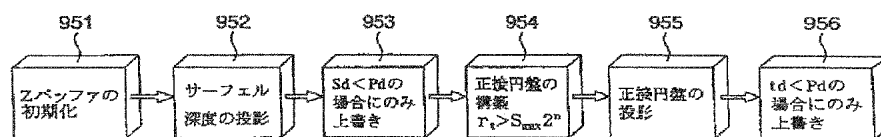
【図6】



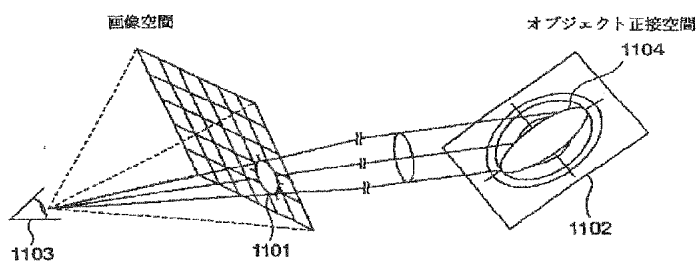
【圖 12】



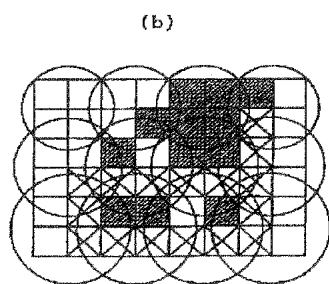
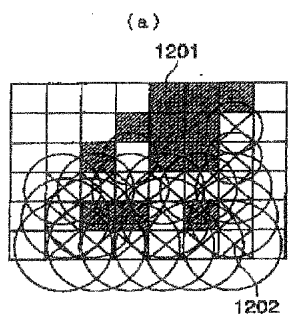
【图 11】



【図13】



【図14】



フロントページの続き

(71)出願人 597067574
201 BROADWAY, CAMBRIDGE,
MASSACHUSETTS
02139, U. S. A.

(72)発明者 ジェロン・ヴァン・パール
オランダ国、1788 ヴェーエル・デン・ヘルダー、
ヴィールバルク 2906
(72)発明者 マルクス・ハー・グロス
スイス国、8610 ウスター、フフスガッセ
2